MARTIN RESEARCH
Microcomputer Division
1825 S. Halsted St. Chicago, Ill. 60608
312 829-6932

6620 (1)

```
        MIKE2 MONITOR - VERSION 13
        --------------------------


        AUTHOR:
        -------

                MARK A. CONDIC, III
                DEPARTMENT OF COMPUTER SCIENCE
                ILLINOIS INSTITUTE OF TECHNOLOGY
                CHICAGO, IL  60616


        ONE PAGE MONITOR PROGRAM TO CONTROL KEYBOARD

        INTEL$                      . DEFINE 8008 ASSEMBLER STUFF

        EQU'S FOR MONITOR PROGRAM

RAM     EQU     010*256             . RAM IS CURRENTLY PAGE 010
SAVE    EQU     013*256+0370        . SAVEAREA FOR STATUS ON INTERRUPTS
RAMPGM  EQU     RAM                 . PROGRAM STARTS AT PAGE 'RAM'
PROM1   EQU     001*256             . NEXT PROM IS PAGE ONE

        SUBROUTINES USEFUL TO OTHER PROGRAMS:

        LOC     NAME                DESCRIPTION
        ---     ----                -----------

        001     UNKEY               WAIT UNTIL ALL KEYS UNPRESSED (DEBOUNCE)
        010     RST010              TIMED LOOP OF 6,667 MSEC
        020     RST020              PROGRAM BREAKPOINT
        030     RST030              READ NEXT CHARACTER FROM CALC KEYBOARD
        040     RST040              WAIT 1 SEC., OR A CALC KEY IS DEPRESSED
        054     INHL                INCREMENT L (AND H) REGISTER
        060     RST060              SEE IF ANY KEYS DEPRESSED (BUT DON'T READ)
        073     STATUS              SAVE 8008 STATUS AND REGISTERS
        151     REST                RESTORE SAVED 8008 STATUS AND REGISTERS
        322     PARTHL              DISPLAY H AND L REGISTER IN OCTAL
        350     PARTC               DISPLAY REGISTER C IN OCTAL (WITH FILLS)
        363     EXP                 EXPLODE REG C INTO OCTAL DIGITS (FOR OUTPUT)


        MIKE2 - MONITOR FUNCTIONS
        -------------------------

        KEY     HEX                 FUNCTION
        ---     ---                 --------

        9       9                   OFF TO PROM1

        =       A                   WRITE DISPLAY ACCUM INTO MEMORY
                                    CLEAR DISPLAY ACCUM
                                    INCREMENT L AND H REGISTERS
                                    DISPLAY CONTENTS OF MEMORY

        -       B                   INCREMENT L AND H REGISTERS
                                    CLEAR DISPLAY ACCUM
                                    DISPLAY CONTENTS OF MEMORY

        *       C                   LOAD H REGISTER FROM DISPLAY ACCUM

        /       D                   LOAD L REGISTER FROM DISPLAY ACCUM
                                    DISPLAY CONTENTS OF MEMORY

        .       E                   OFF TO RAM

        C       F                   CONTINUE WHERE INTERRUPTED
                                    (WITH UPDATED REGISTERS IF CHANGED)

        0-7     0-7                 INSERT DIGIT INTO DIPLAY ACCUM
```

```
                        .
                        .
                        .       REGISTER AND STATUS SAVEAREA:
                        .       ----------------------------
                        .
                        .       PAGE    LOC             REGISTER
                        .       ----    ---             --------
                        .
                        .       013     370             A
                        .       013     371             B
                        .       013     372             C
                        .       013     373             D
                        .       013     374             E
                        .       013     375             STATUS
                        .       013     376             H
                        .       013     377             L


                        .         ----------
                        .
                        .       LOCATION ZERO CONTAINS THE INTERRUPT JAM INSTRUCTION
                        .
000000  025     RST000  RST     020             . OFF TO SAVE STATUS ON INTERRUPT
                        .
                        .       HERE TO WAIT UNTIL ALL KEYS 'UN' PRESSED
                        .
000001  015     UNKEY   RST     010             . WAIT 6.667 MSEC
000002  065             RST     060             . ANY KEYS DEPRESSED?
000003  120             JFS     UNKEY           . LOOP IF ONE (OR MORE) KEYS DEPRESSED
000004  001
000005  000
000006  301             LAB                     . COPY 'B' INTO 'A' ALSO (SEE RST 030)
000007  015             RST     010             . WAIT 6.667 MSEC
                                                . MERGE TO WAIT ANOTHER 6.667 MS AND RETURN
                        .
                        .       LOOP TO WAIT 6.667 MSEC
                        .
                        .       RES     010-$           . ALIGN FOR RST 010
000010  026     RST010  LCI     0147            . TIME CONSTANT OF 6.667 MSEC
000011  147
000012  021     TLOOP   DCC                     . DECREMENT COUNTER
000013  110             JFZ     TLOOP           . LOOP UNTIL COUNTER ZERO
000014  012
000015  000
000016  007             RET                     . RETURN
                        .
                        .       INTERRUPT OR PROGRAM BREAKPOINT COMES HERE
                        .
000017                  RES     020-$           . ALIGN FOR RST 020
000020  106     RST020  CAL     STATUS          . SAVE CPU STATUS
000021  073
000022  000
000023  046             LEI     000             . CLEAR DISPLAY ACCUM REGISTER
000024  000
000025  104             JMP     DISPLY          . OFF TO MONITOR (DISPLAY MEMORY)
000026  261
000027  000
                        .
                        .       HERE TO READ NEXT KEY INTO REGISTERS 'A', AND 'B'
                        .       DESTOYS A,B,C
                        .
                        .       RES     030-$           . ALIGN FOR RST 030
000030  065     RST030  RST     060             . ANY KEYS DEPRESSED?
000031  160             JTS     RST030          . LOOP UNTIL KEY DEPRESSED
000032  030
000033  000
000034  312             LBC                     . COPY DEPRESSED KEY NUMBER
000035  104             JMP     UNKEY           . NOW MERGE TO WAIT FOR KEY TO BE 'UN' PRESS
000036  001
000037  000
```

```
                        .
                        .       HERE TO APPEAR TO PAUSE TO OPERATOR -->
                        .       UNLESS A KEY IS DEPRESSED
                        .
                                RES     040-$           . ALIGN FOR RST 040
000040  016     RST040  LBI     0200            . OUTSIDE TIME CONSTANT FOR 1.2 SEC
000041  200
                        .
                        .       INNER LOOP OF A FEW MILLESECONDS (6.667) OR UNTIL KEY DEPRESSED
                        .
000042  015     RST042  RST     010             . WAIT 6.667 MSEC
000043  065             RST     060             . ANY KEYS DEPRESSED?
000044  023             RFS                     . RETURN NOW IF SO.
000045  011             DCB                     . LOOP UNTIL
000046  120             JFS     RST042          . TIME CONSTANT = 0377
000047  042
000050  000
000051  007             RET                     . RETURN WITH CONDITION SIGN = 1
                        .
                        .       HERE TO INCREMENT H AND L REGISTER
                        .
000052  046     XINHL   LEI     000             . CLEAR E REG, MERGE INHL
000053  000
000054  060     INHL    INL                     . INCREMENT L REGISTER
000055  013             RFZ                     . RETURN IF NO OVERFLOW
000056  050             INH                     . INCREMENT H REGISTER
000057  007             RET                     . RETURN
                        .
                        .       SUBPROGRAM TO SEE IF ANY KEYS ARE DEPRESSED
                        .
                        .       CONDITION CODE SIGN=1 IF NO KEYS DEPRESSED
                        .       CONDITION CODE SIGN=0 IF ONE KEY DEPRESSED
                        .       REGISTER 'C' IS KEY DEPRESSED
                        .       DESTROYS A,B,C
                        .
                                RES     060-$           . ALIGN FOR RST 060
000060  026     RST060  LCI     020             . NUMBER OF KEYS TO CHECK
000061  020
                        .
                        .       HERE TO LOOK FOR A KEY DEPRESSED
                        .
000062  021     RST062  DCC                     . DECREMENT KEY LOOP COUNTER
000063  063             RTS                     . RETURN IF NO KEYS DEPRESSED
000064  302             LAC                     . COPY NEXT KEY NUMBER TO CHECK
000065  115             INP     006             . SEE IF THIS KEY IS DEPRESSED
000066  012             RRC                     . IS KEY DEPRESSED?
000067  063             RFC                     . RETURN IF KEY DEPRESSED
000070  104             JMP     RST062          . SEE IF MORE KEYS TO DO
000071  062
000072  000
                        .
                        .       SUBROUTINE TO SAVE 8008 STATUS AND REGISTERS
                        .
000073  113     STATUS  INP     005             . SAVE A REGISTER IN EXTERNAL 2 WORD QUEUE
000074  305             LAH                     . COPY H REGISTER
000075  113             INP     005             . SAVE H REGISTER IN EXTERNAL 2 WORD QUEUE
000076  306             LAL                     . COPY L REGISTER
000077  056     SAV     LHI     H(SAVE)         . GENERATE PAGE ADDRESS
000100  013
000101  066             LLI     L(SAVE+7)       . AND ENDING OFFSET OF SAVEAREA
000102  377
000103  370             LMA                     . SAVE L REGISTER
                        .
                        .       NOW SAVE STATUS BITS
                        .
000104  006             LAI     000             . ASSUME ZERO FLAG
000105  000
000106  150             JTZ     SAVC            . IF ZERO, THEN CHECK CARRY AND PARITY
000107  120
000110  000
000111  066             LAI     060             . ASSUME NO SIGN BIT
000112  060
000113  120             JFS     SAVC            . WE WERE RIGHT, SO CHECK CARRY AND PARITY
000114  120
000115  000
```

```
000116  066                     LAI     0300            . SET SIGN BIT FLAG
000117  300
000120  032     SAVC    RAR                             . SAVE CARRY
000121  170                     JTP     SAVW            . JUMP IF PARITY EVEN
000122  126
000123  000
000124  054                     XRI     001             . SAVE PARITY ODD FLAG
000125  001
000126  061     SAVW    DCL                             . POINT TO NEXT SAVEAREA WORD
000127  113                     INP     005             . SAVE STATUS, GET A , BUT WE WANT H
000130  113                     INP     005             . SAVE A, GET H REGISTER
000131  370                     LMA                     . SAVE H REGISTER IN MEMORY
000132  061                     DCL                     . POINT TO NEXT SAVEAREA WORD
000133  113                     INP     005             . GET STATUS
000134  370                     LMA                     . SAVE STATUS
000135  061                     DCL                     . POINT TO NEXT SAVEAREA WORD
000136  374                     LME                     . SAVE E REGISTER
000137  061                     DCL                     . POINT TO NEXT SAVEAREA WORD
000140  373                     LMD                     . SAVE D REGISTER
000141  061                     DCL                     . POINT TO NEXT SAVEAREA WORD
000142  372                     LMC                     . SAVE C REGISTER
000143  061                     DCL                     . POINT TO NEXT SAVEAREA WORD
000144  371                     LMB                     . SAVE B REGISTER
000145  061                     DCL                     . POINT TO NEXT SAVEAREA WORD
000146  113                     INP     005             . RESTORE A REGISTER
000147  370                     LMA                     . SAVE A REGISTER
000150  007                     RET                     . HAVING SAVED EVERYTHING RETURN

                        .
                        .       SUBROUTINE TO RESTORE REGISTERS AND STATUS SAVED BY SAVE
                        .
        151     PGMF    EQU     $                       . 'F' --> CONTINUE WHERE INTERRUPTED
000151  056     REST    LHI     H(SAVE)                 . GENERATE PAGE AND WORD
000152  013
000153  066             LLI     L(SAVE+7)               . OFFSET OF ENDING LOCATION OF SAVEAREA
000154  377
000155  307                     LAM                     . GET L
000156  113                     INP     005             . SAVE IT FOR NOW
000157  061                     DCL                     . POINT TO NEXT SAVEAREA WORD
000160  307                     LAM                     . GET H
000161  113                     INP     005             . SAVE IT TOO, FOR NOW
000162  061                     DCL                     . POINT TO NEXT SAVEAREA WORD
000163  307                     LAM                     . RESTORE FLAG BITS
000164  061                     DCL                     . POINT TO NEXT SAVEAREA WORD
000165  347                     LEM                     . RESTORE E REGISTER
000166  061                     DCL                     . POINT TO NEXT SAVEAREA WORD
000167  337                     LDM                     . RESTORE D REGISTER
000170  061                     DCL                     . POINT TO NEXT SAVEAREA WORD
000171  327                     LCM                     . RESTORE C REGISTER
000172  061                     DCL                     . POINT TO NEXT SAVEAREA WORD
000173  317                     LBM                     . RESTORE B REGISTER
000174  061                     DCL                     . POINT TO NEXT SAVEAREA WORD
000175  200                     ADA                     . RESTORE FLAGS BITS
000176  307                     LAM                     . GET A REGISTER
000177  113                     INP     005             . SAVE A, GET L
000200  360                     LLA                     . RETORE L REGISTER
000201  113                     INP     005             . GET H REGISTER
000202  350                     LHA                     . RETORE H REGISTER
000203  113                     INP     005             . RESTORE A REGISTER
000204  007                     RET                     . RETURN


                        .
                        .       A = WRITE NEXT LOCATION , THEN INC H AND L REGISTER
                        .
000205  024     PGMA    SUI     013                     . FORM 'B' ORIGIN
000206  013
000207  100                     JFC     PGMB            . NOT 'A' SO TRY 'B'
000210  214
000211  000
000212  374                     LME                     . WRITE MEMORY FROM DISPLAY REGISTER
000213  250                     XRA                     . CLEAR 'A' FOR MERGE WITH INC H AND L
```

```
                        .
                        .       B = INCREMENT H AND L REGISTER, THEN READ
                        .
000214  150     PGMB    JTZ     INCHL           . 'B' --> INCREMENT, THEN READ
000215  244
000216  000
                        .
                        .       E = JUMP TO RAM,  F = CONTINUE AFTER INTERRUPT
                        .
000217  024     PGME    SUI     003             . FORM 'E' ORIGIN
000220  003
000221  150             JTZ     RAMPGM          . 'E' --> OFF TO PROGRAM IN RAM
000222  000
000223  010
000224  100             JFC     PGMF            . 'F' --> CONTINUE LAST EXECUTION (RESTART)
000225  151
000226  000
                        .
                        .       D = LOAD THE 'L' REG, DISPLAY NEXT MEMORY LOCATION
                        .
000227  306     PGMD    LAL                     . COPY L REGISTER IN CASE WE ARE WRONG
000230  364             LLE                     . LOAD L REGISTER FROM DISPLAY ACCUM
000231  342             LEC                     . CLEAR DISPLAY ACCUM
000232  170             JTP     DISMEM          . 'D' --> LOAD 'L' AND DISPLAY MEMORY
000233  247
000234  000
                        .
                        .       C = LOAD THE 'H' REGISTER
                        .
000235  356     PGMC    LHL                     . LOAD 'H' REGISTER FROM OLD DISPLAY ACCUM
000236  360             LLA                     . RESTORE SAVED L REGISTER SINCE WRONG
000237  036             LDI     0356            . FILL CHARACTER
000240  356
000241  104             JMP     OUTACM          . MERGE TO OUTPUT ACCUM
000242  313
000243  000


                        .
                        .       HERE TO INC H AND L REGISTER
                        .
000244  106     INCHL   CAL     XINHL           . CLEAR E REG, INC H AND L REGISTERS
000245  052
000246  000
                        .
                        .       HERE TO DISPLAY CONTENTS OF MEMORY
                        .
000247  036     DISMEM  LDI     0314            . FILL CHARACTER OF 'CC' (FOR CONTENTS OF)
000250  314
000251  327             LCM                     . LOAD CONTENTS OF MEMORY
000252  106             CAL     PARTC           . OUTPUT C REGISTER
000253  350
000254  000
000255  045             RST     040             . PAUSE AND SEE IF ANOTHER KEY DEPRESSED
000256  120             JFS     FNC             . YES --> OFF TO PERFORM THAT FUNCTION
000257  270
000260  001
                        .
                        .       DISPLAY THE H AND L REGISTERS
                        .
000261  105     DISPLY  CAL     PARTHL          . DISPLAY 'H' AND 'L' REGISTER
000262  322
000263  000
000264  045             RST     040             . PAUSE AND SEE IF ANOTHER KEY DEPRESSED
000265  160             JTS     DISMEM          . NO --> DISPLAY CONTENTS OF MEMORY
000266  247
000267  000
                        .
                        .       NOTE FALL THRU TO NEXT PAGE
```

```
                        .
                        .       HERE TO GET NEXT FUNCTION
                        .
000270  035     FNC     RST     030             . READ THE NEXT KEY FROM KEYBOARD
000271  074             CPI     011             . THE DIGITS A - F ?
000272  011
000273  150             JTZ     PROM1           . '9' --> OFF TO PROM1
000274  000
000275  001
000276  100             JFC     PGMA            . YES THEN GO PROCESS SPECIAL CODES
000277  205
000300  000
                        .
                        .       HERE TO INSERT NEXT OCTAL DIGIT INTO DISPLAY ACCUM (REG E)
                        .
000301  304             LAE                     . COPY DISPLAY ACCUM
000302  002             RLC                     . SHIFT OVER
000303  002             RLC                     . THREE BITS
000304  002             RLC                     . FOR NEXT OCTAL DIGIT
000305  044             NDI     0370            . NOW CLEAR BITS FOR NEXT OCTAL DIGIT
000306  370
000307  201             ADB                     . PLUS THE NEXT CHARACTER
000310  340             LEA                     . STORE UPDATED LSB
000311  036             LDI     0252            . FILL CHARACTER 'AA' FOR ACCUM
000312  252
                        .
                        .       HERE TO OUTPUT DISPLAY ACCUM, AND FILL CHARACTER
                        .
000313  324     OUTACM  LCE                     . LEAST SIGNIFICANT PART
000314  106             CAL     PARTC           . OUTPUT THIS INFO
000315  350
000316  000
000317  104             JMP     FNC             . LOOK FOR NEXT FUNCTION
000320  270
000321  000
                        .
                        .       HERE TO OUTPUT 'H' AND 'L' REGISTER TO HEX DISPLAY PORTS
                        .
                        .       DESTROYS A,B,C
                        .
000322  326     PARTHL  LCL                     . LOAD LEAST SIGNIFICANT WORD
000323  106             CAL     EXP             . EXPLODE OCTAL DIGITS
000324  363
000325  000
000326  133             OUT     033             . OUTPUT RIGHT 2 DIGITS
000327  305             LAH                     . LOAD MOST SIGNIFICANT WORD
                        .
                        .       REGISTER A CONTAINS MOST SIGNIFICANT PART, AND
                        .       REGISTER 'B' THE REMAINDER FROM LEAST SIGNIFICANT PART
                        .
000330  012     PART2   RRC                     . MULTIPLY
000331  012             RRC                     . BY 2**3
000332  012             RRC                     . TO MOVE 3 BITS
000333  320             LCA                     . SAVE FOR H REGISTER
000334  012             RRC                     . ONE MORE BIT
000335  044             NDI     0160            . EXTRACT -210----
000336  160
000337  201             ORB                     . FORM   -210--76
000340  135             OUT     016             . OUTPUT H DIGIT AND REST OF L DIGITS
000341  106             CAL     EXP             . EXPLODE MODIFIED H REGISTER
000342  363
000343  000
000344  044             NDI     077             . KILL HIGH ORDER BITS
000345  077
000346  137             OUT     017             . OUTPUT THE HIGH 2 H DIGITS
000347  007             RET                     . RETURN
```

```
                        .
                        .       HERE TO OUTPUT BINARY REGISTER 'C' AND FILL CHARS IN REG 'D'
                        .
000350  106     PARTC   CAL     EXP             . EXPLODE BINARY REGISTER C
000351  363
000352  000
000353  133             OUT     015             . OUTPUT RIGHT MOST TWO DIGITS
000354  303             LAD                     . COPY FILL CHARACTERS
000355  137             OUT     017             . OUTPUT FILL CHARACTERS
000356  044             NDI     0360            . CLEAR RIGHT DIGIT
000357  360
000360  261             ORB                     . FORM FILL CHAR, OCTAL DIGIT
000361  135             OUT     016             . OUTPUT IT
000362  007             RET                     . RETURN

                        .
                        .       EXPLODE BINARY REG C INTO PRINTABLE OCTAL FOR HEX OUTPUT PORTS
                        .
                        .       DESTROYS A,B
                        .
                        .       NOTE THAT '-' DENOTES ZERO BITS
                        .
000363  302     EXP     LAC                     . COPY BINARY REGISTER
000364  002             RLC                     . POSITION BITS
000365  002             RLC                     . 7 AND 6
000366  044             NDI     003             . FORM ------76
000367  003
000370  310             LBA                     . SAVE IT
000371  302             LAC                     . COPY BINARY REGISTER
000372  044             NDI     0370            . ISOLATE XX543---
000373  370
000374  202             ADC                     . FORM X543-210
000375  044             NDI     0177            . FORM -543-210
000376  177
000377  007             RET                     . RETURN
```